

IT IS CLAIMED:

1. In a nonvolatile memory organized into a plurality of blocks, each block being a plurality of memory units that are erasable together, each memory unit for storing a logical unit of data, a method of storing and updating data in said memory comprising:

organizing data into a plurality of logical groups, each logical group partitioned into a plurality of logical units of data; each logical unit storable in a memory unit of a block;

storing all logical units of a given logical group among memory units of a first block according to a first order;

storing a series of updated logical units of said given logical group among memory units of a second block dedicated to said given logical group according to a second order, until a predetermined closure condition exists that closes said second block to storing further updates; and

responsive to said second block being closed, either:

replacing said first block with said second block when said second and first orders are similar, or

replacing said first block with a third block when said second and first orders are not similar, said third block having been consolidated according to said first order with a latest version of each logical unit of said given logical group gathered from said first and second blocks.

2. A method as in 1, further comprising:

responsive to said second block being closed and said second block containing distinct logical units less than a predetermined number, compacting to a fourth block a latest version of each logical unit from said second block; and

replacing said second block with said fourth block as a dedicated block for storing updates of said given logical group.

3. A method as in 2, wherein said predetermined number of distinct logical units is half of the number of logical units in said given logical group.

4. A method as in 1, wherein said predetermined condition for closure is when said second block is full.

5. A method as in 1, wherein said predetermined condition for closure is when said second block has hitherto been storing the series of updates in an order similar to said first order but for the current update in the series, and said second block is at least half filled.

6. A method as in 5, wherein said second block is closed by having been consolidated according to said first order with a latest version of each logical unit of said given logical group gathered from said first and second blocks.

7. A method as in 1, wherein:
when said storing a series of updated logical units to said second block has hitherto been in an order similar to said first order but for a gap spanned by an address jump in the current update in the series, further includes:
filling said gap with logical units spanned by the address jump copied from said first block before storing the current update in the series.

8. A method as in 1, wherein:
when said storing a series of updated logical units to said second block has hitherto been in an order similar to said first order but for the current update in the series, further includes:
maintaining an index of updated logical units stored in said second block according to said second order.

9. A method as in 8, wherein said index is maintained in said nonvolatile memory.

10. A method as in 1, wherein said given logical group is one of multiple logical groups concurrently under updating by said method.

11. A method as in 1, wherein said nonvolatile memory has floating gate memory cells.

12. A method as in 1, wherein said nonvolatile memory is flash EEPROM.

13. A method as in 1, wherein said nonvolatile memory is NROM.

14. A method as in 1, wherein said nonvolatile memory is a memory card.

15. A method as in any one of 1-14, wherein said nonvolatile memory has memory cells that each stores one bit of data.

16. A method as in any one of 1-14, wherein said nonvolatile memory has memory cells that each stores more than one bit of data.

17. A nonvolatile memory comprising:

a memory organized into a plurality of blocks, each block being a plurality of memory units that are erasable together, each memory unit for storing a logical unit of data;

a controller for controlling operations of said blocks;

said controller storing all logical units of a given logical group thereof among memory units of a first block according to a first order;

said controller storing a series of updated logical units of said given logical group among memory units of a second block dedicated to said given logical group according to a second order until a predetermined closure condition exists that closes said second block to storing further updates; and

responsive to said second block being closed, said controller either:

replacing said first block with said second block when said second and first orders are similar, or

replacing said first block with a third block when said second and first orders are not similar, said third block having been stored by said controller according to said first order a latest version of each logical unit of said given logical group gathered from said first and second blocks.

18. A nonvolatile memory as in 17, further comprising:
responsive to said second block being closed and said second block containing distinct logical units less than a predetermined number, said controller storing to a fourth block a latest version of each logical unit from said second block; and
replacing said second block with said fourth block as a dedicated block for storing updates of said given logical group.

19. A nonvolatile memory as in 18, wherein said predetermined number of distinct logical units is half of the number of logical units in said given logical group.

20. A nonvolatile memory as in 17, wherein said predetermined condition for closure is when said second block is full.

21. A nonvolatile memory as in 17, wherein said predetermined condition for closure is when said second block has hitherto been storing the series of updates in an order similar to said first order but for the current update in the series, and said second block is at least half filled.

22. A nonvolatile memory as in 21, wherein said second block is closed by having been consolidated according to said first order with a latest version of each logical unit of said given logical group gathered from said first and second blocks.

23. A nonvolatile memory as in 17, wherein:
when said storing a series of updated logical units to said second block has hitherto been in an order similar to said first order but for a gap spanned by an address jump in the current update in the series, further includes:
filling said gap with logical units spanned by the address jump copied from said first block before storing the current update in the series.

24. A nonvolatile memory as in 17, wherein:

when said storing a series of updated logical units to said second block has hitherto been in an order similar to said first order but for the current update in the series, further includes:

maintaining an index of updated logical units stored in said second block according to said second order.

25. A nonvolatile memory as in 24, wherein said index is maintained in said nonvolatile memory.

26. A nonvolatile memory as in 17, wherein said given logical group is one of multiple logical groups concurrently under updating by said method.

27. A nonvolatile memory as in 17, wherein said nonvolatile memory has floating gate memory cells.

28. A nonvolatile memory as in 17, wherein said nonvolatile memory is flash EEPROM.

29. A nonvolatile memory as in 17, wherein said nonvolatile memory is NROM.

30. A nonvolatile memory as in 17, wherein said nonvolatile memory is a memory card.

31. A nonvolatile memory as in any one of 17-30, wherein said nonvolatile memory has memory cells that each stores one bit of data.

32. A nonvolatile memory as in any one of 17-30, wherein said nonvolatile memory has memory cells that each stores more than one bit of data.